

数值计算之美

SHU ZHI JI SUAN ZHI MEI

胡家威

<http://hujiaweibujidao.github.io/>



清华大学逸夫图书馆 · 北京

内 容 简 介

本书是我对数值计算中的若干常见的重要算法及其应用的总结，内容还算比较完整。

本人才疏学浅，再加上时间和精力有限，所以本书不会详细介绍很多的概念，需要读者有一定的基础或者其他的参考书籍，这里推荐参考文献中的几本关于数值计算的教材。

本书只会简单介绍下算法的原理，对于每个算法都会附上我阅读过的较好的参考资料以及算法的实现 (Matlab 或者其他语言)，大部分代码是来源于参考文献 [1] 或者是经过我改编而成的，肯定都是可以直接使用的，需要注意的是由于 Latex 对代码的排版问题，导致中文注释中的英文字符经常出现错位，对于这种情况请读者自行分析，不便之处还望谅解。写下这些内容的目的是让自己理解地更加深刻些，顺便能够作为自己的 HandBook，如有错误之处，还请您指正，本人邮箱地址是：hujiawei090807@gmail.com。

目 录

第三章 矩阵特征值和奇异值的求解	1
3.1 幂法求矩阵特征值	1
3.1.1 基本幂法	1
3.1.2 反幂法和移位幂法	2
3.2 QR 分解的迭代过程求矩阵特征值	3
3.2.1 QR 分解的迭代过程	3
3.2.2 Householder 变换	3
3.2.3 简易版本的 QR 分解	5
3.2.4 实用 QR 分解的有关技术	7
3.3 计算矩阵的奇异值	8
3.3.1 奇异值分解 SVD	8
参考文献	9

第三章 矩阵特征值和奇异值的求解

3.1 幂法求矩阵特征值

3.1.1 基本幂法

若矩阵表示向量空间到自身的变换时，那么特征值很重要。基本幂法可以求解得到矩阵最大的特征值 (主特征值)，参考文献 [3](P124 – 125) 有详细的证明过程。算法过程如下：

- (1) 任取一个长度为 n 的非零列向量 x_i ;
- (2) 用矩阵 a 乘以 x_i ，得到列向量 $x_{i+1} = a * x_i$;
- (3) 标准化列向量 x_{i+1} ，也就是表示成一个因子 (绝对值最大的元素的倒数) 乘以 x_{i+1} ，使得 x_{i+1} 中最大的元素是 1;
- (4) 将 x_{i+1} (不包括上面的因子) 作为 (1) 中的 x_i 重复 (1) 中的操作。这个迭代过程一直进行，直到相邻两个列向量 x_i 和 x_{i+1} 的差异满足一定的条件，这个差异可以用多种方式进行衡量，如果使用无限范数来衡量即是

$$\left\| x_{i+1} - x_i \right\|_{\infty} \leq tol$$

最大的特征值就是当前 x_{i+1} 进行标准化得到的因子。幂法肯定是会收敛的，如果初始列向量和特征向量很接近的话收敛会很快，否则会很慢。幂法只能计算出最大的特征值，并且最大的特征值不能是特征方程的重根，其次，最大特征值必须是实数。

code/powerEigenm.m

```
function y = powerEigen(A,x)
% 使用幂法求解矩阵的最大特征值的简易版本
% A=[4 2 -2; -2 8 1; 2 4 -4];
% x=[1;1;1];
% y=powerEigen(A,x)

tol=0.001;
xn=A*x;%下一个列向量
xp=x;%上一个列向量
while max(abs(xn-xp))>tol
    xp=xn;
    y=max(abs(xn));%最大特征值
    xn=xn./y;
    xn=A*xn;
end
end
```

3.1.2 反幂法和移位幂法

由于矩阵 A 的特征值 λ 的倒数就是矩阵 A^{-1} 的特征值，所以，如果对 A^{-1} 进行同样的幂法过程，那么就可以得到矩阵 A 的最小的特征值，这种方法就叫做反幂法 (详细内容请看参考文献 [3](P128))。

如果得到了矩阵 A 的最大或者最小的特征值，那么使用移位幂法可以得到其他的特征值。它的原理是：假设 $ax = \lambda_1 x$ ，其中 λ_1 是通过幂法求得的矩阵的最大的特征值，那么新的移位矩阵 $[a - \lambda_1 * I]$ 的特征值便是 $\lambda_2 - \lambda_1, \lambda_3 - \lambda_1, \dots, \lambda_n - \lambda_1$ (证明很简单，此处忽略)。移位矩阵的特征向量和原矩阵的特征向量是一样的。如果对移位矩阵也应用基本幂法，得到特征值 α_k ，那么原矩阵的特征值 $\lambda_k = \alpha_k + \lambda_1$ 。不断重复上面的过程 ($k-2$) 次，便可以得到原矩阵的所有的特征值 (详细内容请看参考文献 [3](P129))。

3.2 QR 分解的迭代过程求矩阵特征值

QR 分解是求解中小规模矩阵所有特征值的有效方法，它的基本原理是相似变换 (两个方阵相似的条件是存在可逆矩阵 C , 使得 $A = C^{-1}BC$) 不改变矩阵的特征值和对应的特征向量以及上三角矩阵的特征值就是它的对角线元素。

实 Schur 型矩阵: 矩阵 S 为上三角阵, 且对角块为 1 阶或者 2 阶矩阵。这类矩阵可以进行正交分解: $S = Q^T A Q$, 称为实 Schur 分解, 其中 S 的 1 阶对角块就是 A 的实特征值, 2 阶对角块的特征值是 A 的共轭复特征值。等式 $A = Q S Q^T$ 称为矩阵 A 的实 Schur 分解。

QR 分解的目标是通过一系列正交相似变换将原矩阵转换成一个上三角型的相似矩阵 (即实 Schur 型矩阵), 这样求解特征值就特别简单了。QR 分解可以得到所有的特征值, 但是不能得到特征向量。如果所有的特征值都是实数的话, QR 分解得到的是一个正交阵和一个上三角阵。如果有的特征值是复数的话, 那么得到的是一个正交阵和一个对角线某些位置为 (2×2) 块矩阵的上三角阵。

3.2.1 QR 分解的迭代过程

假设矩阵 A_1 是要求特征值的矩阵, QR 分解的迭代过程如下:

- (1) 对 A_1 进行 QR 分解得到 $A_1 = Q_1 R_1$ 。其中 Q_1 是一个正交阵, R_1 是一个上三角阵。
- (2) 用 R_1 乘以 Q_1 得到 A_2 , 即 $A_2 = R_1 Q_1$ 。由于 $R_1 = Q_1^{-1} A_1$, 则有 $A_2 = Q_1^{-1} A_1 Q_1$, 所以 A_2 和 A_1 是相似矩阵。
- (3) 此时, 对 A_2 进行 QR 分解得到 $A_2 = Q_2 R_2$, 再将得到的 Q_2 和 R_2 反过来相乘, 得到 A_3 , 然后再对 A_3 进行 QR 分解, 不断重复上面的操作直到得到上三角矩阵, 最终的上三角矩阵对角线元素便是矩阵特征值。

3.2.2 Householder 变换

上面的迭代过程中, 每步都要进行一次 QR 分解, 这是通过 Householder 变换实现的。一个 $(n \times n)$ 的 Householder 矩阵 H 具有如下的形式:

$$H = I - \frac{2}{v^T v} v v^T$$

其中, 矩阵 I 是一个 $(n \times n)$ 单位阵 (对角线元素都是 1), 而 v 是一个包含 n 个元素的列向量

$$v = c + \|c\|_2 e$$

向量 c 和 e 是都是包含 n 个元素的列向量, 分别对应要变换的矩阵的某一行和单位阵的某一行, $\|c\|_2$ 是向量 c 的二范数

$$\|c\|_2 = \sqrt{c_1^2 + c_2^2 + c_3^2 + \cdots + c_n^2}$$

矩阵 H 是正交对称阵, 所以 $H = H^T = H^{-1}$ 。Householder 变换实现向量在线性空间中的“镜面反射”, 即 Hx 是向量 x 相对于法向量为 v 的超平面的镜像。 Hx 和 x 关于平面 S 镜像对称。采用 Householder 变换可以将向量 x 中除某一个分量之外的其他分量变成 0, 这是一种消元操作。注意一类小技巧, 不需要显式构造 H 便可以计算 Hx , 如下所示, 只需计算向量 x 和 v^T 的内积, 不用计算矩阵和向量的乘法。(详细内容可以参考文献 [2](P165-P166))

$$Hx = \left(I - \frac{2}{v^T v} v v^T\right)x = x - 2 \frac{v^T x}{v^T v} v$$

将 $(n \times n)$ 矩阵 A 分解为正交阵 Q 和上三角阵 R 的过程需要 $n - 1$ 步。

(1) 向量 c 是矩阵 A 的第一列, 向量 e 是长度为 n 的列向量, 如果 c 的第一个元素是正数的话 e 的第一个元素是 1, 否则是 -1(这么做其实是为了防止大数吃小数现象, 也就是减小误差)。得到 c 和 e 之后, 便可以计算出 v , 然后得到矩阵 H_1 , 如下所示:

$$c = \begin{bmatrix} a_{11} \\ a_{12} \\ \cdots \\ a_{nn} \end{bmatrix} \quad e = \begin{bmatrix} \pm 1 \\ 0 \\ \cdots \\ 0 \end{bmatrix}$$

将 H_1 作用于 A ，此时 $Q_1 = H_1, R_1 = H_1 A$ 。其中 Q_1 是正交阵， R_1 的第一列第一个元素以下的元素都是 0。 R_1 大致如下所示：

$$\begin{pmatrix} R_{11}^{(1)} & R_{12}^{(1)} & \cdots & R_{1n}^{(1)} \\ 0 & R_{22}^{(1)} & \cdots & R_{2n}^{(1)} \\ 0 & R_{32}^{(1)} & \cdots & R_{3n}^{(1)} \\ 0 & R_{42}^{(1)} & \cdots & R_{4n}^{(1)} \end{pmatrix}$$

(2) 这步中向量 c 是矩阵 R_1 的第二列，并且第一个元素设置为 0，向量 e 是长度为 n 的列向量，不过第二个元素是 ± 1 ，如果 c 的第二个元素是正数的话 e 的第二个元素是 1，否则是 -1。得到 c 和 e 之后，便可以得到矩阵 H_2 ，如下所示：

$$c = \begin{bmatrix} 0 \\ R_{12}^{(1)} \\ \cdots \\ R_{n2}^{(1)} \end{bmatrix} \quad e = \begin{bmatrix} 0 \\ \pm 1 \\ \cdots \\ 0 \end{bmatrix}$$

将 H_2 作用于 R_1 ，此时 $Q_2 = Q_1 H_2, R_2 = H_2 R_1$ 。其中 Q_2 是正交阵， R_2 的第二列第二个元素以下的元素都是 0。 R_2 大致如下所示：

$$\begin{pmatrix} R_{11}^{(2)} & R_{12}^{(2)} & \cdots & R_{1n}^{(2)} \\ 0 & R_{22}^{(2)} & \cdots & R_{2n}^{(2)} \\ 0 & 0 & \cdots & R_{3n}^{(2)} \\ 0 & 0 & \cdots & R_{4n}^{(2)} \end{pmatrix}$$

(3) 重复与上面类似的过程，直到第 $(n-1)$ 步，得到 $Q_{n-1} = Q_{n-2} H_{n-1} = H_1 H_2 \cdots H_n$ ，同时 $R_{n-1} = H_{n-1} R_{n-2} = H_{n-1} H_{n-2} \cdots H_2 H_1$ ，此时 R_{n-1} 是一个上三角阵， Q_{n-1} 是一个正交阵，得到矩阵 A 的 QR 分解式为 $A = Q_{n-1} R_{n-1}$ 。

3.2.3 简易版本的 QR 分解

简易版本的 QR 分解的代码如下所示：

code/easyqrnm.m

```

function [Q,R] = easyqrnm(R)
% 简易版本的分解QR
% A=[45 30 -25; 30 -24 68; -25 68 80];
% for i=1:40
%     [q,r]=easyqrnm(A);
%     A=r*q;
% end
% e=diag(A)
% eig(A)

n=size(R);
I=eye(n);
Q=I;
for j=1:n-1
    c=R(:,j);
    c(1:j-1)=0;
    e(1:n,1)=0;%
    if c(j) > 0
        e(j)=1;
    else
        e(j)=-1;
    end
    clen=sqrt(c'*c);
    v=c+clen*e;
    H=I-2/(v'*v) * v*v';%得到矩阵H
    Q=Q*H;%
    R=H*R;%
end

```

3.2.4 实用 QR 分解的有关技术

实际使用的 QR 分解要比上面的简易版本的 QR 分解复杂很多，但是基本思路是一样的，那就是通过一系列的正交相似变换 $B = Q^T A Q$ 将矩阵 A 化为上三角或者对角块阶数很小的分块上三角阵，然后求特征值。实用的技术主要有两个：(1) 将矩阵化简为上 Hessenberg 矩阵；(2) 带原点位移的 QR 算法。

上 Hessenberg 矩阵和上三角矩阵的区别在于，其紧邻主对角线下方的副对角线上的元素不全为 0。可以证明，对上 Hessenberg 矩阵 A_k 进行 QR 算法的一步迭代，得到的 A_{k+1} 仍为上 Hessenberg 矩阵。使用 Householder 变换可以将一般矩阵化为上 Hessenberg 矩阵，总共经过 $(n-2)$ 步正交相似变换。若原始矩阵是实对称阵，那么得到的上 Hessenberg 矩阵是一个对称的三对角阵。对上 Hessenberg 矩阵进行 QR 分解时不需要使用 Householder 变换，只要使用一系列 Givens 旋转变换即可。Givens 旋转变换是一类平面旋转变换，它能够消去给定向量的某一个分量 (使其为 0)，这不同于 Householder 变换消去向量中的多个分量。在处理稀疏向量或者稀疏矩阵时，Givens 旋转更加高效。

将原点位移技术与 QR 分解结合，通过原点位移技术可以改变做 QR 分解的矩阵，一方面它能够提高迭代收敛速度，另一方面也使 QR 分解对更一般的矩阵有效。

该部分的内容比较难，此处不过多介绍，详细内容参考文献 [2](P174-P178)。

3.3 计算矩阵的奇异值

若矩阵表示一个空间到另一个空间的变换，奇异值很重要。求矩阵的奇异值可以利用求矩阵的特征值的方法，最简单的求矩阵 A 奇异值的方法是计算矩阵 $A^T A$ 的特征值，非负特征值的平方根就是特征值 (对称矩阵的一个性质)，但是这种方法的计算误差大。

还有一种求矩阵奇异值的方法是先将矩阵 A 化简成双对角阵 B ，再对 $B^T B$ 执行 (隐式)QR 迭代算法。 $B_k^T B_k$ 在 QR 迭代过程中保持形状不变。

3.3.1 奇异值分解 SVD

这部分内容参考自文献 [4] 中的 Dimensionality Reduction 中的 Singular Value Decomposition。在该网页中可以查看详细例子。

对于 $(n \times d)$ 矩阵 X 的奇异值分解式是： $X = U A V^T$ ，其中 $(n \times n)$ 矩阵 U 和 $(d \times d)$ 矩阵 V 都是正交阵，而 $(n \times d)$ 矩阵 A 是对角阵，对角线上的元素是矩阵 A 的奇异值。矩阵 U 的列向量是矩阵 X 的左奇异向量，矩阵 V 的列向量是矩阵 X 的右奇异向量。矩阵 A 对角线上的奇异值自上到下是从大到小分布的，并且所有的奇异值都大于 0。

要计算矩阵 X 的奇异值分解式，就要计算 $X^T X$ 和 $X X^T$ 的特征值和特征向量。其中，矩阵 $X^T X$ 的特征向量就是矩阵 V 的列，而矩阵 $X X^T$ 的特征向量就是矩阵 U 的列，矩阵 X 的奇异值就是矩阵 $X^T X$ 和 $X X^T$ 的共同非负特征值的平方根。矩阵 X 的奇异值的个数是矩阵 X 的秩，也就是线性无关的行 (或者列) 的数目。

SVD 分解的过程：

- (1) 计算矩阵 $X^T X$ 和矩阵 $X X^T$ 的特征值和特征向量；
- (2) 计算矩阵 $X^T X$ 和矩阵 $X X^T$ 的共同非负特征值的平方根得到奇异值；
- (3) 通过计算的结果得到矩阵 U ，矩阵 V 和矩阵 A 。

参考文献

- [1] Numerical Computing with Matlab. Cleve B. Moler.
中文翻译版本《Matlab 数值计算》，喻文健，机械工业出版社，2006，6
网站资源：
 - (1)[Cleve Moler 撰写的教科书](#)
 - (2)[数值计算交互演示网站](#)
- [2] 数值分析与算法，喻文健，清华大学出版社，2012，1
- [3] Numerical Methods: An introduction with Applications Using Matlab. Amos Gilat. Vish Subramaniam, 2010, 10
- [4] Data Mining Algorithms In R.
网站资源：
[WikiBook: Data Mining Algorithms In R](#)